BAB 4
IMPLEMENTASI DAN PENGUJIAN

4.1 Implementasi

Tahap implementasi merupakan tahap kelanjutan dari kegiatan perancangan sistem. Wujud dari hasil implementasi ini nantinya adalah sebuah sistem yang siap untuk diuji dan digunakan.

4.2 Perangkat Pendukung yang digunakan

Dalam proses pembangunan aplikasi *game* ini, tentunya membutuhkan perangkat keras (*Hardware)* dan perangkat lunak (*Software).* Berikut merupakan penjelasan dari perangkat keras dan perangkat lunak yang digunakan.

4.2.1 Perangkat Keras *(Hardware)*

Perangkat keras yang digunakan selama pembangunan aplikasi ini memiliki spesifikasi sebagai berikut :

Tabel 4.1 Spesifikasi Perangkat Keras yang digunakan

Nama Perangkat	Spesifikasi
Processor	AMD A8-4500M Quad Core
Memory	4 GB DDR3
Hard Disk	SATA 500 GB
Video Card	Radeon HD 8750M, 2GB DDR3
Mouse	Optic 1 Unit
Keyboard	1 Unit
Speaker	Realtek High Audio

4.2.2 Perangkat Lunak *(Software)*

Spesifikasi perangkat lunak yang digunakan dalam pembangunan *game* ini adalah sebagai berikut :

Tabel 4.2 Spesifikasi Perangkat Lunak yang digunakan

Nama Perangkat Lunak	Spesifikasi
Sistem Operasi	Windows 7 Ultimate
Tools Model 3	1. iClone 3DXchange v5.5 Pipeline 2. Blender 2.68a 3. 3ds MAX 2014 4. SketchUp 2013
Tools Pembangun *Game*	Unity 4.1.5f1

4.3 Implementasi Antarmuka

Pada bagian ini akan ditampilkan implementasi antarmuka *Menu* utama, *tutorial*, *about*, *game*, dan misi.

4.3.1 Antarmuka *Menu* Utama

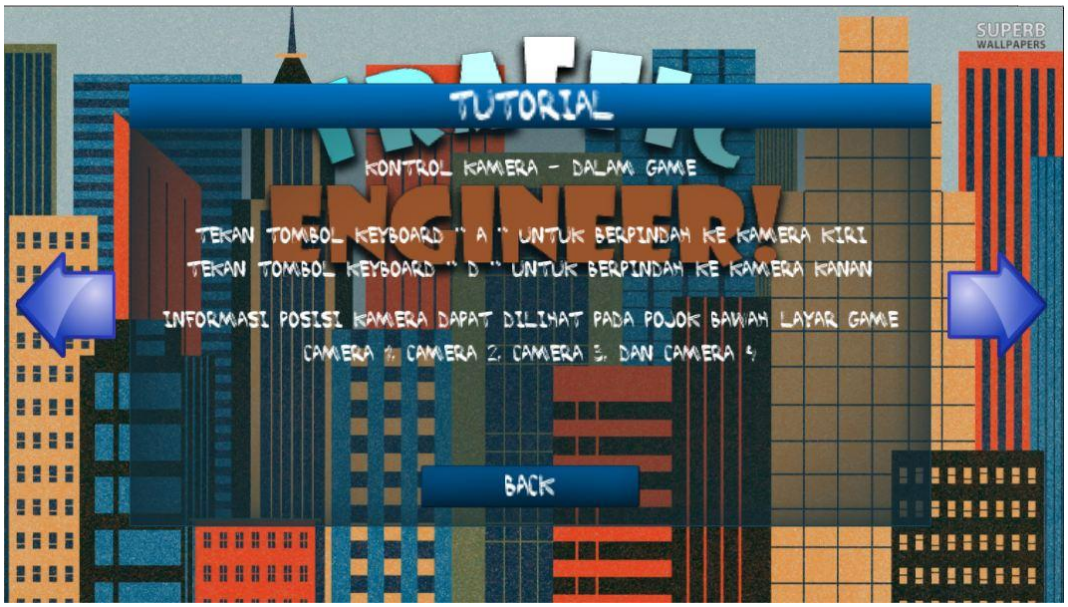
Layar ini berfungsi menampilkan tampilan *Menu* utama atau tampilan awal dari *game* yang dibangun, dimana didalamnya terdapat tiga pilihan *Menu* yaitu play, *tutorial*, *about*, dan *exit.*



Gambar 4.1 Antarmuka *Menu* Utama

4.3.2 Antarmuka *Menu Tutorial*

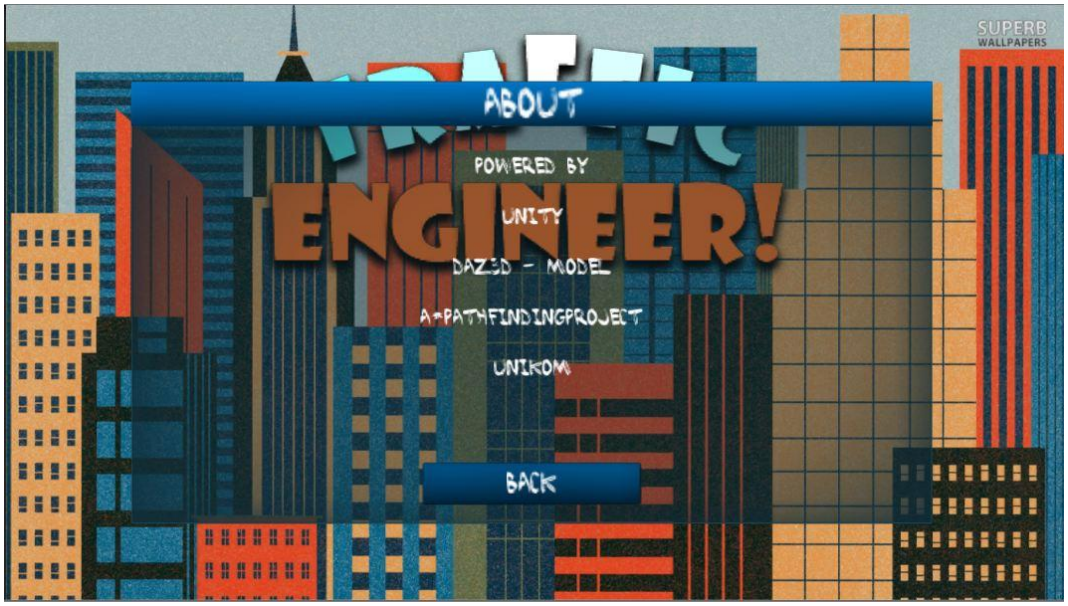
Layar ini berfungsi menampilkan tampilan *Menu Tutorial* dari *game* yang dibangun. Pada antarmuka ini terdapat tiga tombol yaitu *back*, *next*, *previous.*



Gambar 4.2 Antarmuka *Menu* Tutorial

4.3.3 Antarmuka *Menu About*

Layar ini berfungsi menampilkan tampilan *Menu About* dari *game* yang dibangun. Pada antarmuka ini juga terdapat tombol back untuk kembali ke tampilan menu utama.



Gambar 4.3Antarmuka *Menu* About

4.3.4 Antarmuka *Game*

Layar ini berfungsi menampilkan tampilan *game* secara keseluruhan dalam *game* yang dibangun.



Gambar 4.4Antarmuka *Game*

4.3.5 Antarmuka Misi

Layar ini berfungsi menampilkan tampilan informasi misi dalam *game* yang dibangun.

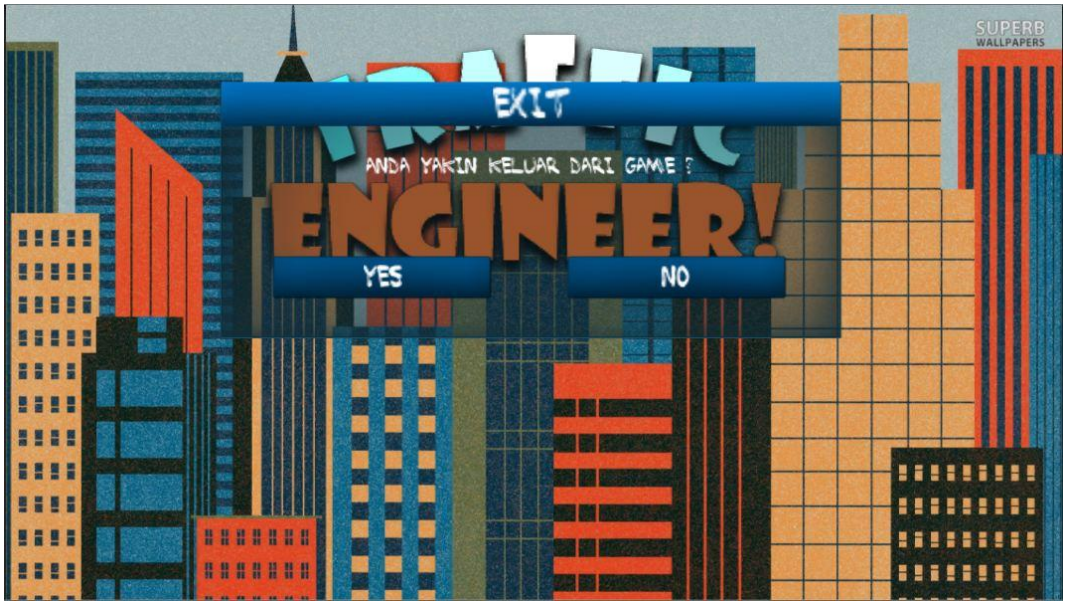


Gambar 4.5 Antarmuka *Menu* Misi

4.4 Implementasi Antarmuka Pesan

Pada bagian ini akan ditampilkan implementasi antarmuka *Menu* utama, *tutorial*, *about*, *game*, dan misi.

4.4.1 Antarmuka Pesan *Exit Game*



Gambar 4.6Antarmuka Pesan Exit *Game*

4.5 Pengujian Sistem

Pengujian bermaksud untuk mengetahui perangkat lunak yang dibuat sudah memenuhi kriteria yang sesuai dengan tujuan perancangan perangkat lunak

4.6 Rencana Pengujian

Pengujian akan dilakukan melalui dua tahap, yaitu pengujian *Alpha* dan pengujian *beta*.

1. Pengujian *Alpha*

   Metode yang digunakan dalam pengujian *Alpha* adalah metode *Black box* yang fokus pada persyaratan fungsional dari perangkat lunak yang dibangun.

2. Pengujian *Beta*

Metode yang digunakan dalam pengujian *beta* merupakan pengujian yang dilakukan secara objektif terhadap pengguna.

4.7 Pengujian *Alpha*

Pengujian dilakukan terhadap aplikasi untuk memastikan bahwa aplikasi berjalan dengan benar sesuai dengan kebutuhan dan tujuan yang diharapkan. Proses pengujian akan menggunakan *Black box Testing*.

4.8 Skenario Pengujian

Skenario pengujian menjelaskan pengujian terhadap sistem yang ada pada *game* yang dibangun. Pengujian dilakukan dengan melakukan tes setiap *use case* dan memungkinkan kesalahan yang terjadi pada setiap proses. Pengujian ini dilakukan secara *Black box* yaitu pengujian dilakukan dengan hanya memperhatikan masukan ke sistem dan keluaran sistem. Skenario pengujian yang akan dilakukan pada aplikasi ini selengkapnya dijelaskan pada tabel 4.3.

Tabel 4.3 Skenario Pengujian Alpha

No.	Komponen yang diuji	Skenario Pengujian	Jenis Pengujian
1	*Menu Tutorial*	Menampilkan Layar *Tutorial*	*Black box*
		Memilih Tombol *Next*	*Black box*
		Memilih Tombol *Previous*	*Black box*
		Memilih Tombol *Back*	*Black box*
2	*Menu About*	Menampilkan Layar *About*	*Black box*
		Memilih *Menu Back*	*Black box*
3	*Menu Exit*	Menampilkan Pesan *Exit*	*Black box*
		Memilih Tombol *Yes*	*Black box*
		Memilih Tombol *No*	*Black box*
4	*Start Game*	Menampilkan Layar Permainan	*Black box*
5	Pengaturan Kamera	Menekan Tombol Keyboard A	*Black box*
		Menekan Tombol Keyboard D	*Black box*

6	Pengaturan Lampu Lalu Lintas	Pilih Lampu Lalu Lintas	*Black box*
7	*Menu* Misi	Pilih Tombol Misi	*Black box*
		Menampilkan Layar Misi	*Black box*
		Pilih Tombol *Back*	*Black box*
8	Pause *Game*	Pilih Tombol Pause	*Black box*
		Menampilkan Pesan Pause	*Black box*
		Pilih Tombol Resume	*Black box*
		Pilih Tombol *Exit*	*Black box*
9	Algoritma A* (A-Star)	Algoritma A* pada permainan	White Box

4.9 Pengujian *Black box*

Berdasarkan rencana pengujian yang telah disusun, maka di dapatkan hasil pengujian yang akan dijelaskan dalam table 4.4.

Tabel 4.4 Hasil Pengujian *Black box – Menu* Tutorial

Menu Tutorial		
Masukan	Hasil yang diharapkan	Hasil pengujian
Klik *Menu Tutorial*	Menampilkan layar *tutorial*	[√] diterima [ ] ditolak
Klik Tombol *Next*	Menampilkan *tutorial* kedua	[√] diterima [ ] ditolak
Klik Tombol *Previous*	Menampilkan *tutorial* kesatu	[√] diterima [ ] ditolak
Klik Tombol *Back*	Menampilkan layar *Menu* utama	[√] diterima [ ] ditolak

Tabel 4.5 Hasil Pengujian *Black box – Menu* About

Menu About		
Masukan	Hasil yang diharapkan	Hasil pengujian
Klik *Menu About*	Menampilkan layar *about*	[√] diterima [ ] ditolak
Klik Tombol *Back*	Menampilkan layar *Menu* utama	[√] diterima [ ] ditolak

Tabel 4.6 Hasil Pengujian *Black box – Menu* Exit

Menu Exit		
Masukan	Hasil yang diharapkan	Hasil pengujian
Klik *Menu Exit*	Menampilkan pesan *exit*	[√] diterima [ ] ditolak
Klik Tombol *Yes*	Keluar dari *game*	[√] diterima [ ] ditolak
Klik Tombol *No*	Menampilkan layar *Menu* utama	[√] diterima [ ] ditolak

Tabel 4.7 Hasil Pengujian *Black box – Start Game*

Start Game		
Masukan	Hasil yang diharapkan	Hasil pengujian
Klik *Menu Start Game*	Menampilkan layar permainan	[√] diterima [ ] ditolak

Tabel 4.8 Hasil Pengujian *Black box* – Pengaturan Kamera

Pengaturan Kamera		
Masukan	Hasil yang diharapkan	Hasil pengujian
Tekan Tombol Keyboard A	Berpindah kamera ke kiri	[√] diterima [ ] ditolak
Tekan Tombol Keyboard D	Berpindah kamera ke kanan	[√] diterima [ ] ditolak

Tabel 4.9 Hasil Pengujian *Black box* – Pengatura Lampu

Pengaturan Lampu Lalu Lintas		
Masukan	Hasil yang diharapkan	Hasil pengujian
Klik Lampu Lalu Lintas	Merubah warna menjadi kuning	[√] diterima [ ] ditolak

Tabel 4.10 Hasil Pengujian *Black box – Menu* Misi

Menu Misi		
Masukan	Hasil yang diharapkan	Hasil pengujian
Klik *Menu* Misi	Menampilkan layar misi	[√] diterima [ ] ditolak
Klik *Menu Back*	Kembali ke layar *game*	[√] diterima [ ] ditolak

Tabel 4.11 Hasil Pengujian *Black box* – Pause *Game*

Pause *Game*		
Masukan	Hasil yang diharapkan	Hasil pengujian
Klik *Menu* Pause	Menampilkan pesan pause	[√] diterima [ ] ditolak
Klik *Menu* Resume	Kembali ke layar *game*	[√] diterima [ ] ditolak
Klik *Menu Exit*	Kembali ke *Menu* utama	[√] diterima [ ] ditolak

4.10 Kesimpulan

Dari hasil pengujian *Black box* yang dilakukan, maka didapat kesimpulan bahwa seluruh fungsional yang terdapat dalam game sudah berjalan dengan benar dan terpenuhi tanpa terjadinya kesalahan dalam proses menjalankanya.

4.11 Pengujian *White Box*

Berikut ini adalah kasus menguji perangkat lunak yang telah dibangun menggunakan metode *white-box* berdasarkan tabel 4.3. Pengujian *white-box* menggunakan *flow graph* yang digunakan untuk menggambarkan alur dari algoritma dan *grap matrix* yang digunakan untuk menggenerasi *flow graph*. Adapun pengujian *white-box* adalah sebagai berikut :

1. *Source Code*

1	//Teruskan pencarian selama tidak terjadi eror dan node akhir belum ditemukan while (CompleteState == PathCompleteState.NotCalculated) {

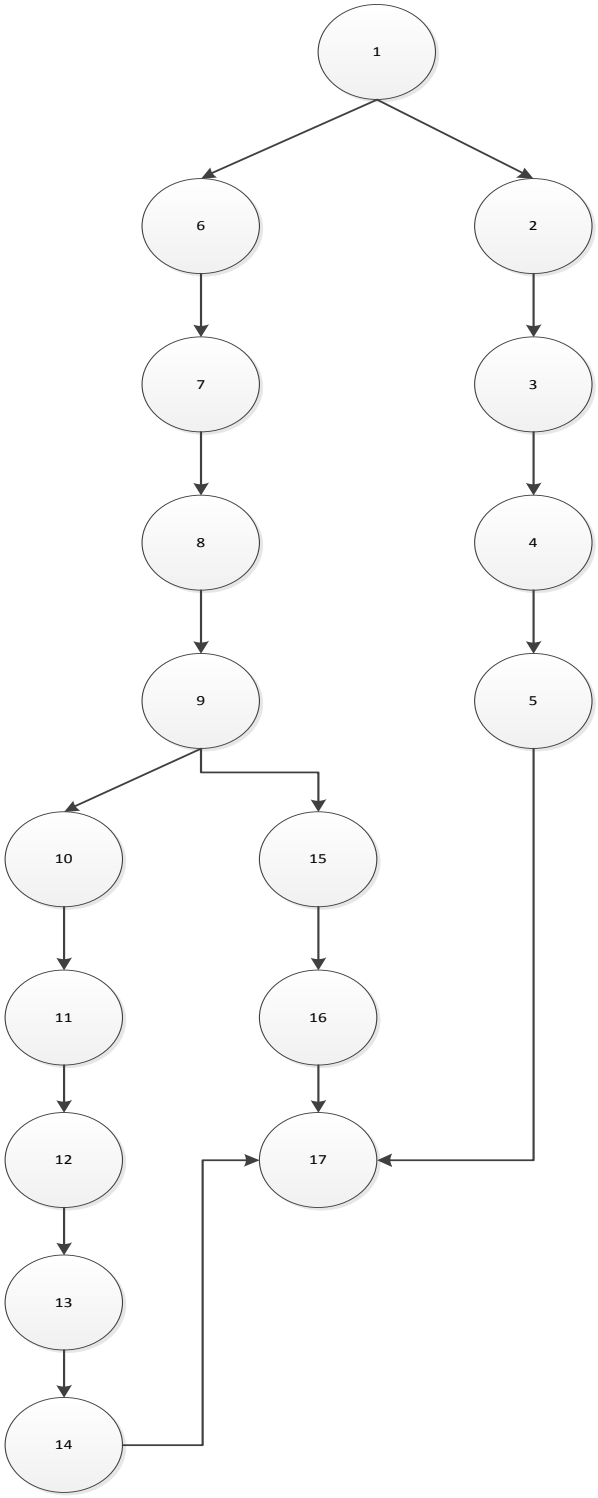
```
     //Close atau tutup current node, jika current node sama dengan node tujuan maka pencarian jalur selesai
2    if (currentR.node == endNode) {
3    CompleteState = PathCompleteState.Complete;
4    break;
5    }
6    if (currentR.h < partialBestTarget.h) {
7    partialBestTarget = currentR;
8    }

     //Ulangi pada semua node tetangga yang dapat dilewati dan masukan kedalam open list
9    currentR.node.Open (runData,currentR, hTarget,this);

     //apakah terdapat node untuk dilakukan pencarian?
10   if (runData.open.numberOfItems <= 1) {
11   Error ();
12   LogError ("No open points, whole area searched");
13   return;
14   }
15   //pilih node dengan nilai F score terkecil dan hapus node tersebut dari open list
16   currentR = runData.open.Remove ();
17   }
```

Gambar 4.7 *Source Code*

2. *Flow Graph*



Gambar 4.8 Flow Graph Algoritma A*

Keterangan :

 = Menggambarkan kondisi

 = Menggambarkan aksi

3. *Cylomatic Complexity* V(G)

V(G) = E – N + 2

V(G) = 18 – 17 + 2

V(G) = 3

Di mana:

E = jumlah edge pada grafik alir

N = jumlah *no*de pada grafik alir

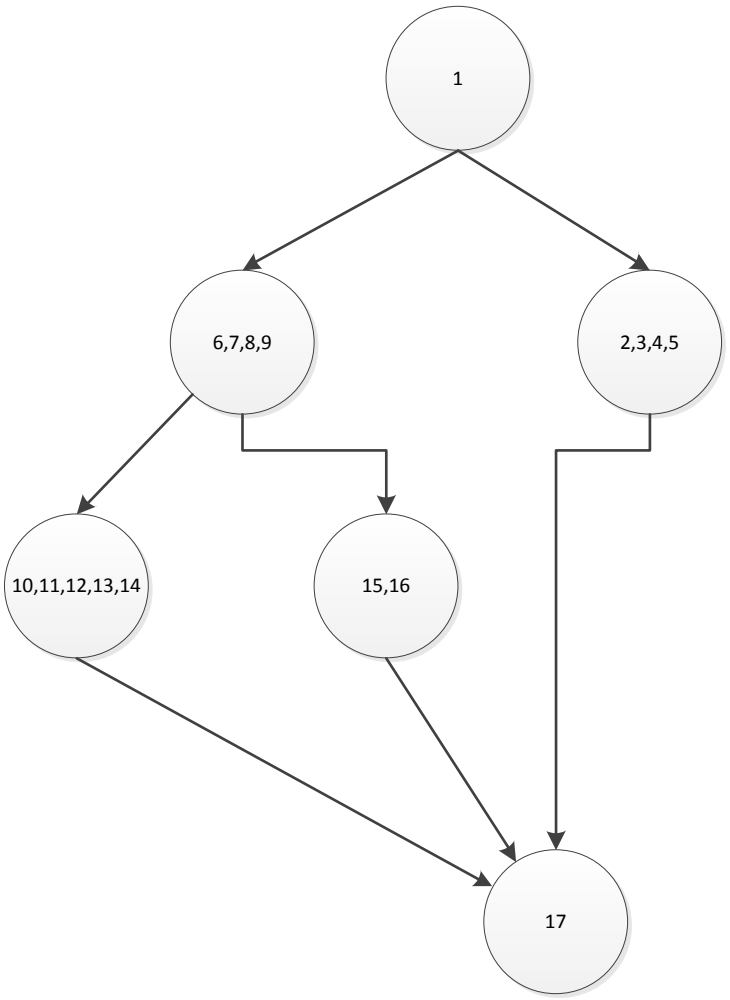
4. *Independent Path*

Path 1 = 1-2-3-4-5-17

Path 2 = 1-6-7-8-9-10-11-12-13-14-17

Path 3 = 1-6-7-8-9-15-16-17

Maka Gambar 4.8 disederhanakan berdasarkan kondisi atau simpul sebagai berikut:



Gambar 4.9 Penyederhanaan Flow Graph Algoritma A*

5. ***Graph Matrices***

Tabel 4.12 Hasil Pengujian *Black box*

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	Hasil
1		1				1												1
2			1															0
3				1														0
4					1													0
5																	1	0
6							1											0
7								1										0

8									1								
9										1					1			1
10											1							0
11												1						0
12													1					0
13														1				0
14																	1	0
15																1		0
16																	1	0
17																	1	0
Hasil																		2

V(G) = Jumlah Graph Matrices + 1

V(G) = 2 + 1

V(G) = 3

6. Kesimpulan

Dari hasil pengujian algoritma A* yang dilakukan, maka didapat kesimpulan bahwa seluruh *node* yang ada pada setiap *independent path* telah dikerjakan, serta tidak terjadi pengulangan tak terhingga.

4.12 Kesimpulan Pengujian *Alpha*

Dari hasil pengujian yang dilakukan, dapat disimpulkan bahwa sistem sudah dapat berjalan sesuai dengan yang diharapkan beserta proses penanganan kesalahannya. Jadi, secara fungsional sistem sudah dapat menghasilkan keluaran yang diharapkan.

4.13 Pengujian *Beta*

Pengujian *beta* merupakan pengujian yang dilakukan secara objektif, dimana dilakukan pengujian secara langsung terhadap pengguna dengan

menggunakan kuesioner mengenai kepuasan pengguna atas aplikasi yang telah dibangun. Adapun metode penilaian pengujian yang digunakan adalah metode kuantitatif berdasarkan data dari pengguna.

Kuesioner disebarkan menggunakan teknik *Simple Random Sampling* dimana anggota *sample* dari populasi dilakukan secara acak tanpa memperhatikan strata yang ada dalam populasi. Kuesioner disebarkan kepada 30 orang dari umur 11 tahun ke atas, dari hasil kuesioner tersebut akan dilakukan perhitungan agar dapat diambil kesimpulan terhadap penilaian penerapan sistem yang telah dibangun. Kuesioner terdiri dari 6 pernyataan dengan menggunakan skala 1 sampai 5 (skala likert).

Untuk mencari nilai persentase dari masing-masing jawaban kuesioner digunakan rumus skala likert sebagai berikut :

P = (S/Skorideal) x 100%

Keterangan :

P = Nilai persentase yang dicari

S = Jumlah frekuensi dikalikan dengan nilai yang di tetapkan jawaban

Skorideal = Nilai tertinggi dikalikan dengan jumlah sampel

Berikut ini adalah hasil persentase masing-masing jawaban yang sudah dihitung nilainya dengan menggunakan rumus diatas.

1. Apakah anda setuju simulasi pengaturan sistem lampu lalu lintas sudah mirip dengan keadaan aslinya?

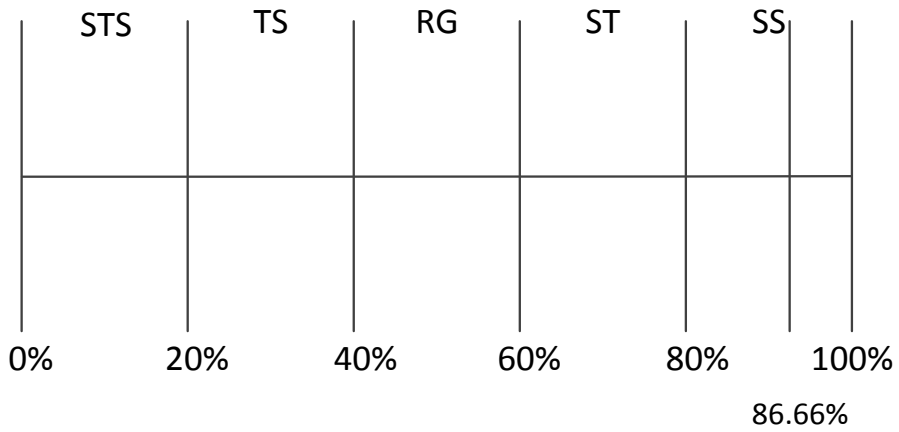
Tabel 4.13 Perhitungan persentase pertanyaan ke-1

Pertanyaan	Keterangan	Skor	Frekuensi	S
1	Sangat Setuju	5	15	75
	Setuju	4	10	40
	Ragu-Ragu	3	5	15
	Tidak Setuju	2	0	0

	Sangat Tidak Setuju	1	0	0
Jumlah			30	130

P = (130/150)x100% = 86.66%

Dapat disimpulkan bahwa penilaian terhadap simulasi pengaturan sistem lampu lalu lintas pada *game* yang dibangun adalah 86.66% dari yang diharapkan 100%, atau dapat dikategorikan sebagai sangat setuju, untuk lebih jelasnya dapat dilihat pada Gambar 4.10.



Gambar 4.10 Interpretasi Skor Pertanyaan ke-1

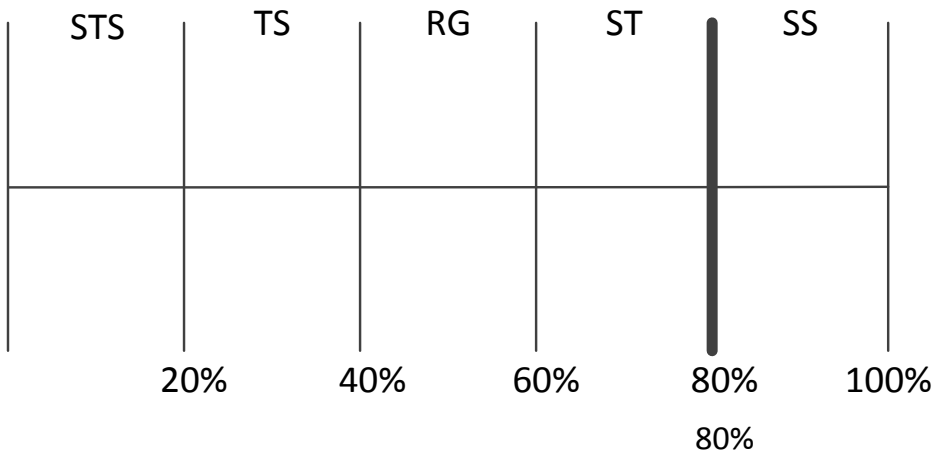
2. Apakah anda setuju simulasi sifat pengguna jalan (jam padat) sudah mirip dengan keadaan aslinya?

Tabel 4.14 Perhitungan persentase pertanyaan ke-2

Pertanyaan	Keterangan	Skor	Frekuensi	S
2	Sangat Setuju	5	10	50
	Setuju	4	11	44
	Ragu-Ragu	3	8	24
	Tidak Setuju	2	1	2
	Sangat Tidak Setuju	1	0	0
Jumlah			30	120

P = (120/150)x100% = 80%

Dapat disimpulkan bahwa penilaian terhadap simulasi sifat pengguna jalan (jam padat) pada *game* yang dibangun adalah 80% dari yang diharapkan 100%, atau dapat dikategorikan sebagai sangat setuju, untuk lebih jelasnya dapat dilihat pada Gambar 4.11.



Gambar 4.11 Interpretasi Skor Pertanyaan ke-2

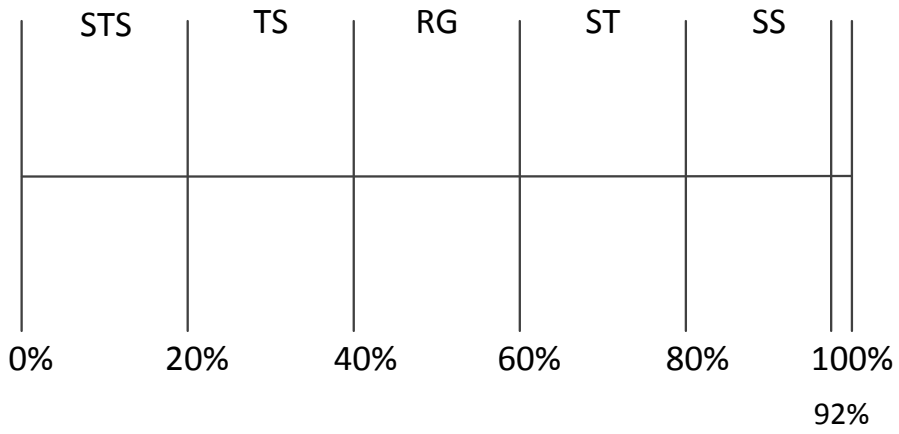
3. Apakah anda setuju *NPC* mobil pada *game* yang dibangun sulit untuk ditebak arah tujuanya?

Tabel 4.15 Perhitungan persentase pertanyaan ke-3

Pertanyaan	Keterangan	Skor	Frekuensi	S
3	Sangat Setuju	5	13	65
	Setuju	4	12	48
	Ragu-Ragu	3	5	25
	Tidak Setuju	2	0	4
	Sangat Tidak Setuju	1	0	0
Jumlah			30	138

P = (138/150)x100% = 92%

Dapat disimpulkan bahwa penilaian terhadap *NPC* mobil pada *game* yang dibangun sulit untuk ditebak arah tujuanya adalah 92% dari yang diharapkan 100%, atau dapat dikategorikan sebagai sangat setuju, untuk lebih jelasnya dapat dilihat pada Gambar 4.12.



Gambar 4.12 Interpretasi Skor Pertanyaan ke-3

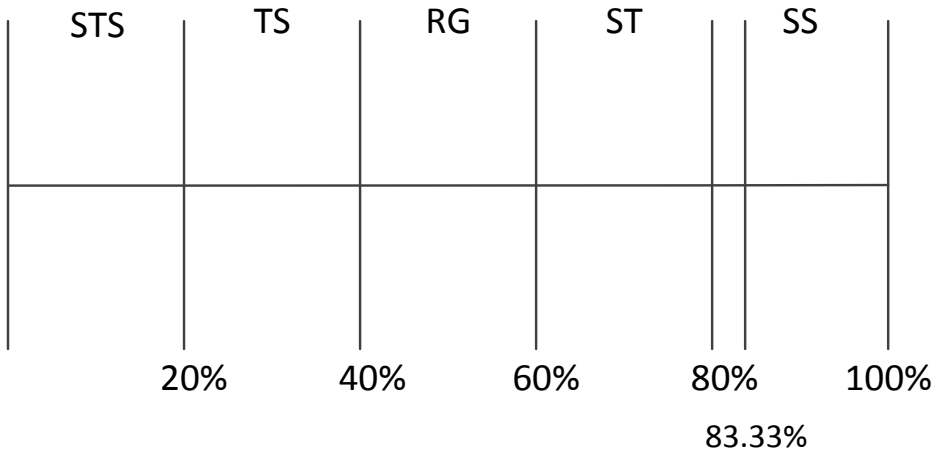
4. Apakah anda setuju dengan adanya tantangan keadaan darurat menambah variasi tantangan baru pada *game* yang bertemakan pengendalian lampu lalu lintas?

Tabel 4.16 Perhitungan persentase pertanyaan ke-4

Pertanyaan	Keterangan	Skor	Frekuensi	S
4	Sangat Setuju	5	10	50
	Setuju	4	15	60
	Ragu-Ragu	3	5	15
	Tidak Setuju	2	0	0
	Sangat Tidak Setuju	1	0	0
Jumlah			30	125

P = (125/150)x100% = 83.33%

Dapat disimpulkan bahwa penilaian terhadap adanya tantangan keadaan darurat menambah variasi tantangan baru pada *game* yang bertemakan pengendalian lampu lalu lintas adalah 83.33% dari yang diharapkan 100%, atau dapat dikategorikan sebagai sangat setuju, untuk lebih jelasnya dapat dilihat pada Gambar 4.13.



Gambar 4.13 Interpretasi Skor Pertanyaan ke-4

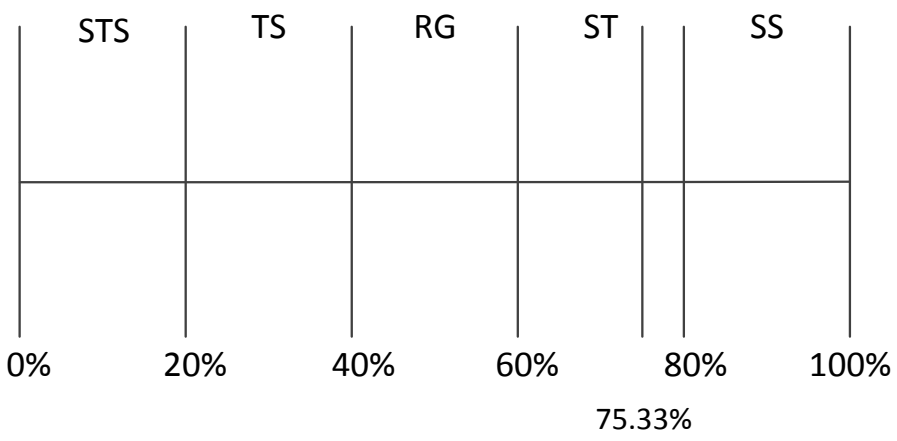
5. Apakah anda setuju *game* yang dibangun menarik untuk dimainkan?

Tabel 4.17 Perhitungan persentase pertanyaan ke-5

Pertanyaan	Keterangan	Skor	Frekuensi	S
5	Sangat Setuju	5	7	35
	Setuju	4	12	48
	Ragu-Ragu	3	8	24
	Tidak Setuju	2	3	6
	Sangat Tidak Setuju	1	0	0
Jumlah			30	113

P = (113/150)x100% = 75.33%

Dapat disimpulkan bahwa penilaian terhadap *game* yang dibangun menarik untuk dimainkan adalah 75.33% dari yang diharapkan 100%, atau dapat dikategorikan sebagai setuju, untuk lebih jelasnya dapat dilihat pada Gambar 4.14.



Gambar 4.14 Interpretasi Skor Pertanyaan ke-5